Overriding subsuming rules

Philippe Besnard, Eric Grégoire, Sébastien Ramon

Overriding subsuming rules

Ph. Besnard [a], É. Grégoire [b,*], S. Ramon [b]

[a] IRIT CNRS UMR 5505, 118 route de Narbonne, F-31062 Toulouse, France
[b] Université Lille – Nord de France, Artois CRIL CNRS UMR 8188, rue Jean Souvraz SP18, F-62307 Lens, France

ABSTRACT

This paper is concerned with intelligent agents that are able to perform nonmonotonic reasoning, not only *with*, but also *about* general rules with exceptions. More precisely, the focus is on enriching a knowledge base Γ with a general rule that is subsumed by other rules already there. Such a problem is important because evolving knowledge needs not follow logic as it is well-known from e.g., the belief revision paradigm. However, belief revision is mainly concerned with the case that the extra information logically conflicts with Γ. Otherwise, the extra knowledge is simply doomed to extend Γ with no change altogether. The problem here is different and may require a change in Γ even though no inconsistency arises. The idea is that when a rule is to be added, it might need to override any rule that subsumes it: preemption must take place. A formalism dedicated to reasoning with and about rules with exceptions is introduced. An approach to dealing with preemption over such rules is then developed. Interestingly, it leads us to introduce several implicants concepts for rules that are possibly defeasible.

1. Introduction

Assume a knowledge base Γ contains the rule *If the switch is on then the light is on*. When *If the switch is on and the lamp bulb is ok then the light is on* needs to be introduced inside Γ, it seems natural to require this new rule to preempt the older rule: it is no longer enough to know that the switch is on to be able to conclude that the light is on, it must additionally be the case that Γ yields the information that the lamp bulb is ok. First, let us observe that a monotonic logic cannot capture such dynamics of reasoning by simply adding the new rule. According to monotonicity, any conclusion drawn from a given set of premises can still be inferred whatever additional premises happen to supplement this set. In such a logic, the statement *the light is on* (concluded from the former rule and the statement *the switch is on*) is still concluded even though the second rule is added, and, worse yet, regardless of any information stating that the lamp bulb is broken. Also, the usual approaches to belief revision following the seminal work in [1] fail to address this issue directly because they make the new information to be set-theoretically unioned with Γ in case no inconsistency arises. In the paper, it is shown how solving the problem can be done through specific *contraction* steps followed by an *expansion* one, as those operations are called in the belief revision research area. Let us stress that moving to a nonmonotonic formalism where exceptions to rules depend on consistency checks like adding *If the switch is on and if it can be consistently assumed that the lamp bulb is ok, then the light is on* does not change the problem.

Technically, the problem can be described as follows: Given a set Γ of formulas and a rule $\mathcal{R}$, what changes should Γ undergo so as to infer $\mathcal{R}$ but not to infer any $\mathcal{R}'$ subsuming $\mathcal{R}$? In symbols, where $\Gamma^\star$ stands for Γ after these changes have taken place,

$$\Gamma^\star \mathrel{|\!\!\sim} \mathcal{R} \quad \text{and} \quad \Gamma^\star \mathrel{|\!\!\not\sim} \mathcal{R}'$$

* Corresponding author.
E-mail addresses: besnard@irit.fr (Ph. Besnard), gregoire@cril.univ-artois.fr (É. Grégoire), ramon@cril.univ-artois.fr (S. Ramon).

Clearly, the problem first requires several matters to be settled. First, the syntax for rules (in which $\mathcal{R}, \mathcal{R}', \ldots$ are expressed) is to be defined. Second, an inference relation (denoted $\mid\!\sim$) allowing rules to be handled needs to be settled. Third, a concept of implicant for rules expressing what does $\mathcal{R}'$ subsuming $\mathcal{R}$ mean needs to be proposed, before an approach to solve the above preemption issue can be defined.

Accordingly, the paper is organized as follows. First, we analyze situations where a new belief requires the subsuming ones to be expelled (but no means is provided to automatically identify such cases —it is not the aim of this paper). In the following Section, a general formalism for representing rules with exceptions is introduced with the aim of encompassing various logic-based approaches allowing such rules, including default reasoning. Section 4 introduces useful inference tools to reason about such rules, while Section 5 connects the tools with default logic. In Section 6, a useful X-derivation concept is proposed, allowing both plain formulas and rules to be inferred under the possible assumption of additional formulas or other rules with exceptions. Sections 7 and 8 investigate concepts of implicant for rules with exceptions. The approach to the preemption issue is then developed in Section 9, based on the X-derivation and the latter implicant concepts. Finally, some avenues for future research are provided in the conclusion.

Throughout the paper, the following notations are used: $\neg$, $\vee$, $\wedge$ and $\supset$ denote the classical negation, disjunction, conjunction and material implication connectives, respectively. When Ω is a set of formulas, $Cn(\Omega)$ denotes the deductive closure of Ω under a given logic, of which $\Vdash$ denotes the consequence relationship,$\perp$ denotes absurdity, and $\top$ denotes any tautology.

2. About new beliefs that require subsuming beliefs to be expelled

The focus in this paper is on rational agents that are equipped with sound and complete logical inference mechanisms about beliefs. As modeled in the so-called AGM framework [1] that gave rise to the belief revision research area, the natural situation where a rational agent is handling a new belief that is not conflicting with its previous ones, requires the belief to be merely added to its set of beliefs.

As illustrated in the introduction, such a very simple schema is not universal and does not cover all possible situations. In some circumstances, a new belief requires some former ones to be expelled even though inconsistency does not threaten.

In this paper, the focus is on new non-conflicting beliefs that are intended to *prevail* over preexisting ones. Prevailing means here that the new belief must be adopted and other beliefs that logically subsume it must be expelled. In the lamp example, *If the switch is on then the light is on* subsumes (i.e., has for strict consequence) *If the switch is on and the lamp is ok then the light is on* and must be expelled so that the new rule prevails. Thus, this study only concerns situations where a new piece of information is given and when this new one must prevail over a pre-existing set of premises.

We believe that this occurs frequently in real-life. A typical situation is when we refine our knowledge and beliefs so that we need to adopt a *more precise* rule and block *less precise* pre-existing ones, as in the lamp example. From a logical point of view, a more precise information can be a strict deductive consequence of a more general one. Hence, the need to get rid of the more general ones.

Let us stress that when we want a new rule *if A and B then C* to prevail, this does not entail that *not B* is a counter-example for deriving *C* from *A*. Actually, *C* can perhaps also be obtained from *A* is true and *B* is false. For example, assume we refine a medical diagnosis system. We might want to assert the rule *If symptom1 and has-been-in-Africa then disease1* and want it to prevail. So we need to reject *If symptom1 then disease1* and adopt the new (more precise) rule. This does not require *not has-been-in-Africa* to be a counter-example from deriving *disease1* from *symptom1*. *Disease1* could perhaps also be derived when we have *symptom1* and *has-been-in-Amazonia*, according to another rule.

Even more importantly, the need for a new piece of information to prevail over the preexisting subsuming premises is actually neither restricted to the (intuitive) concept of “more precise" rules or “more precise" information. Actually, it need not even be restricted to rules (and thus counter-examples to rules), nor to their syntax, but exists for any kind of formulas. For example, assume that the pre-existing beliefs contain *John is in his office or John is at home*. The need to block subsumption also occurs when we want the new, in some sense more accurate, piece of information *John is in his office or John is at home or John is in his club* to prevail. Interestingly, the language that we are going to develop to model rules also allows to represent formulas that do not appear as rules from an epistemological point of view. The treatment allowing new beliefs to prevail applies to them, too.

Some words of caution are needed here since we are using full and complete logical mechanisms. In standard logic, any literal (namely, any possibly negated elementary fact) entails any rule having the literal as a consequent: *Light is on* entails any possible rule having *light is on* as the consequent. Accordingly, when we enforce a rule that must prevail, we might need to get rid of elementary beliefs. In the paper, we always assume that every belief in a set Γ of premises can be expelled when a new belief needs to prevail. On the one hand, applying the technique might thus require to leave outside Γ formulas that must not be expelled in such a process. On the other hand, in many problems involving the analysis of interacting rules, such an analysis must be conducted without case-specific concrete data. For example, in rule-based expert systems, the rule-base contains generic rules whereas the working memory contains data related to a specific case and temporary conclusions. In such a specific context, our technique to make rules prevail must consider the rule-base only.

Interestingly, the approach that we present in this paper can appear as a combination of (forms of) multiple contraction and expansion operators, which have been studied in the standard Boolean logic setting for belief revision (see [5] for a survey on multiple contraction operators). In this respect, this paper is, for one of its facets, a generalization of some first results about blocking subsumption in the standard Boolean case that were presented in [3] and for which rationality postulates are proposed in [2]. In this paper, the preemption issue is investigated within the setting of a family of non-monotonic logics allowing forms of default assumptions. The handling of complex reasonings involving rules with exceptions leads us to elaborate reasoning paradigms that are not captured by existing (generalized) AGM-like frameworks.

3. PEC rules

These last three decades, many research efforts in the Artificial Intelligence community have been devoted to the logic-based formalization of various forms of reasoning, especially defeasible ones, giving rise to large families of nonmonotonic logics. Noticeably, some of the most popular tools to handle forms of defeasible reasoning remain rules with exceptions, e.g., in the form of defaults [6]. They permit an inference system to jump to default conclusions and to withdraw them when new information shows that these conclusions now lead to inconsistency. Usually, such a default rule is based on logical formulas, that is, expressions of a formal language upon which an inference system (no matter how poor or rich) models some kind of reasoning. The goal of this paper is not to introduce a new nonmonotonic logic in this respect. Instead, we settle on a generic framework that allows the various types of knowledge involved in rules with exceptions to be represented in a single uniform way. Instantiating the framework to a given traditional nonmonotonic logic might require additional constraints on the representation and inference formalisms.

Let us first concentrate on rules with exceptions under consistency tests. Our leading example is then expressed as *If the switch is on and if it can be consistently assumed that the lamp bulb is ok, then the light is on* and consists of three parts: its premises, its exceptions, and its conclusions. Accordingly, we thus aim at representing rules with exceptions in a uniform way within a unified setting that is, among other things, meant to be general enough as to encompass e.g. default logic while allowing us to instantiate it to other logical formalisms. It should also allow the representation of both monotonic knowledge and rules involving consistency checks.

Given a logical language, a PEC rule (for Premises-Exceptions-Conclusions) is a triple consisting of three sets of formulas. First, the premises, which are the necessary conditions for *this* rule to apply. Then, the exceptions, which call for consistency tests. Finally, the conclusions, which list the claims that can be made whenever the rule applies.

Definition 1 (*PEC-rule*)**.** A PEC rule is a triple $\mathcal{R} = (\mathcal{P}, \mathcal{E}, \mathcal{C})$ where $\mathcal{P} = \{\rho_1, \ldots, \rho_k\}$ and $\mathcal{C} = \{\varsigma_1, \ldots, \varsigma_n\}$ are consistent sets of formulas and $\mathcal{E} = \{\epsilon_1, \ldots, \epsilon_m\}$ is a set of non-tautological formulas.

In the definition, we impose no constraint on the language: there may, or may not, be connectives such as negation, conjunction, disjunction and the like. There may even be no connective at all. However, an underlying inference relation $\Vdash$ must be available. Of course, this means that the logical formalism used must have a form of tautology (please note the subtlety here: this does not mean that the logical formalism used must have tautologies). In the rest of the paper, we assume however the availability of the connectives of standard Boolean logic.

A PEC rule can be interpreted in different ways, depending on how its set of premises and its set of conclusions are captured logically (presumably, conjunctively or disjunctively). In the sequel, without loss of generality, we consider only unary PEC rules, that is, PEC rules whose sets of premises and sets of conclusions are singletons (the involved formulas having been assembled through the concerned conjunctive or disjunctive paradigm). Abusing the notation in order to improve readability, we often omit curly brackets for these singletons.

Definition 2 (*Unary PEC rule*)**.** A PEC rule $\mathcal{R} = (\mathcal{P}, \mathcal{E}, \mathcal{C})$ is unary iff $\mathcal{P}$ and $\mathcal{C}$ are singleton sets.

Example 1. The PEC rule $(switch\_on, \{\neg lamp\_bulb\_ok\}, light\_on)$ is an encoding of the rule with exception "*If the switch is on and, consistently assuming that the lamp bulb is ok, then the light is on*".

Example 2. The PEC rule $(\{switch\_on, lamp\_bulb\_ok\}, \emptyset, light\_on)$ is an encoding of a similar rule where the impossibility to derive e.g. "*lamp_bulb_ok*" can block the inference of "*light_on*". In this respect, "$\neg$*lamp_bulb_ok*" would be an exception to the rule, which is however not to be included in the set of exceptions of the PEC rule, since it is not a consistency-based exception.

Example 3. The PEC rule $(\top, \emptyset, light\_on)$ is an encoding of the fact *"the light is on"*.

Example 4. The PEC rules $(switch\_on, \emptyset, light\_on)$, $(switch\_on \supset light\_on, \emptyset, \emptyset)$ and $(\emptyset, \emptyset, switch\_on \supset light\_on)$ are various encodings of the exception-free rule *"If the switch is on then the light is on"*.

As can be seen in the previous examples, exceptions to a rule that are supposed to be derived in the monotonic fragment of the logic (vs. consistency checks) are not included in the set of exceptions in the PEC rule, which is devoted to exceptions based on consistency checks. Indeed, we shall adopt a semantical view of formulas of the monotonic fragment of the logic and e.g. allow for a clausal representation. Accordingly, the premise and conclusion status can be interchanged through modus tollens and no information is given to interpret a literal as playing the specific role of an exception to a rule in a clausal representation. Thus, the word “exception" will be reserved in this paper to “exception based on consistency check". The various possible encodings of knowledge between premises and conclusions is similar to e.g. the well-known difference in default logic between defaults with prerequisites and the corresponding prerequisite-free defaults [cf 4].

Also, $\Vdash$ is assumed to admit $\top$ to represent effectively some formula.

4. Reasoning with and about PEC rules

Let us define a concept of a derivation for the very general language of PEC rules. Interestingly, it will not only allow us to handle both monotonic and defeasible rules in the same setting, but it will also allow us to derive both of them.

A word of warning: in the following, $\vdash$ does not represent an inference relation. $\vdash \alpha$ (resp. $\nvdash \alpha$) means that α has (resp. has not) the status “inferred” *within the derivation*. Also, “not inferred within the derivation” does not mean “whose negated form cannot be inferred using the inferred formulas occurring in the derivation” (which is a weaker notion, clearly uninteresting). *A word of terminology:* $\vdash \alpha$ and $\nvdash \alpha$ are said to be *signed formulas*. Most naturally, $\vdash \gamma$ (resp. $\nvdash \gamma$) is said to be positive (resp. negative).

Definition 3 (*Derivation*)**.** Let Γ be a set of PEC rules and $\aleph = (\rho, \{\epsilon_1, \ldots, \epsilon_n\}, \varsigma)$ one PEC rule. A derivation of $\aleph$ from Γ is a tree T whose nodes are standard signed Boolean formulas s.t.:

1. for all leaves of the form $\vdash \alpha$,
   - either $(\top, \emptyset, \alpha) \in \Gamma$,
   - or $(\alpha_1, \emptyset, \alpha_2) \in \Gamma$ and $\alpha = \alpha_1 \supset \alpha_2$,
   - or $\alpha = \rho$,
2. for all leaves of the form $\nvdash \beta$,
   $\beta \notin Cn(\{\gamma \text{ s.t. } (\top, \emptyset, \gamma) \in \Gamma\} \cup \{\gamma_1 \supset \gamma_2 \text{ s.t. } (\gamma_1, \emptyset, \gamma_2) \in \Gamma\} \cup \{\alpha \text{ s.t.} \vdash \alpha \text{ is a node of } T\})$,
3. if $\nvdash \beta$ is a node then it is a leaf,
4. every node $\vdash \alpha$ that is not a leaf has as parents,
   - either a tuple $(\vdash \alpha_1, \ldots, \vdash \alpha_k)$. In this case, $\alpha \in Cn(\{\alpha_1, \ldots, \alpha_k\})$.
   - or a tuple $(\vdash \alpha_1, \nvdash \beta_1, \ldots, \nvdash \beta_m)$ where $m \geq 1$. In this case, $(\alpha_1, \{\beta_1, \ldots, \beta_m\}, \alpha) \in \Gamma$.
5. $\rho \in \{\alpha \text{ s.t.} \vdash \alpha \text{ is a leaf of } T\} \cup \{\top\}$, $\{\epsilon_1, \ldots, \epsilon_n\} = \{\beta \text{ s.t. } \nvdash \beta \text{ is a node of } T\}$, $\vdash \varsigma$ is the root of T.

We note $\Gamma \mid\!\sim^{\{\epsilon_1, \ldots, \epsilon_n\}} \aleph$ and, when $\{\epsilon_1, \ldots, \epsilon_n\}$ is empty, $\Gamma \mid\!\sim \aleph$.

Let us provide some intuitions and examples. First, consider the simple case of standard Boolean logic: items 2 and 3 are ineffective because there are no negative nodes, while items 4 and 5 get simpler for the same reason. Items 1 & 2 require a positive node to correspond either with a standard Boolean formula encoded inside a PEC rule of Γ, or with the premise ρ of the PEC rule $\aleph$ derived by the tree. This latter alternative translates the idea that a derivation allows a rule to be derived through a construction that can assume that the premise of the rule is established. Item 4 gets simpler: only standard deduction can take place in the tree. Indeed, the tree is then reduced to a standard derivation tree from Boolean logic. Item 5 defines the three components of the $\aleph$ rule derived by the tree.

Example 5. Let $\Gamma = \{(a, \emptyset, b), (\top, \emptyset, a), (b, \emptyset, c)\}$. The tree from Fig. 1 is a derivation of $(\top, \emptyset, c)$ from Γ (please note that the first part of item 5 is satisfied by $\rho \in \{\top\}$). Note that this tree is also a derivation of $(a, \emptyset, c)$, of $(a \supset b, \emptyset, c)$ and $(b \supset c, \emptyset, c)$ from Γ.

Example 6. Let $\Gamma = \{(a, \emptyset, b), (b, \emptyset, c)\}$. The tree of Fig. 1 is a derivation of $(a, \emptyset, c)$ from Γ (please note that a plays the role of an additional hypothesis). The tree of Fig. 1 is also a derivation of $(a, \emptyset, c)$ from $\Gamma \cup \{(\top, \emptyset, a)\}$.

Example 7. Let $\Gamma = \{(\top, \emptyset, a), (b, \emptyset, c)\}$. The tree from Fig. 1 is a derivation of $(a \supset b, \emptyset, c)$ from Γ. Note that ρ plays the role of an additional hypothesis. This tree is also a derivation of $(a \supset b, \emptyset, c)$ from $\Gamma \cup \{(\top, \emptyset, \neg a)\}$, although the $\vdash a$ node in the tree is inconsistent with the $\neg a$ fact encoded through $(\top, \emptyset, \neg a)$.

$$\dfrac{\dfrac{\vdash a \quad \vdash a \supset b}{\vdash b} \qquad \vdash b \supset c}{\vdash c}$$

Fig. 1. Tree from Examples 5–7.

$$\frac{\vdash \neg a \quad \vdash a}{\vdash c}$$

Fig. 2. Tree from Example 8.

$$\frac{\dfrac{\vdash a \wedge b \quad \nvdash d \quad \nvdash e}{\vdash f} \qquad \vdash f \supset c}{\vdash c}$$

Fig. 3. Tree from Examples 9 and 10.

$$\frac{\dfrac{\dfrac{\vdash a \wedge b \quad \nvdash d \quad \nvdash e}{\vdash f} \qquad \vdash f \supset d}{\vdash d} \qquad \nvdash \neg c}{\vdash c}$$

Fig. 4. Tree from Example 11.

Example 8. Let $\Gamma = \{(\top, \emptyset, a)\}$. The tree from Fig. 2 is a derivation of $(\neg a, \emptyset, c)$ from Γ. Note that ρ plays the role of an additional hypothesis.

When some PEC rules in Γ have a non-empty set of exceptions, derivation trees may capture reasoning under some proviso(s) (meaning that there are possible exceptions). Item 2 guarantees reasoning to be consistent in the sense that exception-free information from Γ (that may, or may not, occur as positive nodes) does not yield exceptions whose absence is required for the reasoning developed to be acceptable (cf Example 10 with *Cn* being classical logic and Example 11 with *Cn* being an arbitrary logic). This needs not prevent trivialization (in which case only derivations with no negative node may exist). Item 3 indicates that consistency statements occur as hypotheses, they are not inferred. Item 4 makes sure that each node, if not a leaf, is inferred from exception-free information and/or consistency hypotheses. Only inference steps from *Cn* and rules (with exceptions) in Γ may apply. Lastly, item 5 specifies what components the PEC rule derived consists of:

- Its conclusion is the root of the derivation tree.
- Its exceptions exhaust all consistency hypotheses occurring in the derivation tree (cf Example 9).
- Its premise, if not $\top$, either amounts to some exception-free statement represented by a rule from Γ, or it is an extra formula that plays the role of an additional hypothesis in the reasoning (cf Example 9).

Example 9. Let $\Gamma = \{(a \wedge b, \{d, e\}, f), (f, \emptyset, c)\}$.The tree from Fig. 3 is a derivation of $(a \wedge b, \{d, e\}, c)$ from Γ (please note that ρ is $a \wedge b$ that plays the role of an additional hypothesis and that $\{d, e\}$ exhausts all negative nodes of the derivation tree). The tree from Fig. 3 is not a derivation of $(a \wedge b, \{d, e, g\}, c)$ from Γ (item 5 in the definition of a derivation fails because g is listed as an exception of the derived PEC rule but $\nvdash g$ is not a node of the derivation tree). The tree from Fig. 3 is not a derivation of $(b, \{d, e\}, c)$ from Γ (here, item 5 is failed for a different reason: the purported ρ is b but $\vdash b$ is not a leaf of the derivation tree).

Example 10. Let $\Gamma = \{(a \wedge b, \{d, e\}, f), (f, \emptyset, c), (\top, \emptyset, \neg f)\}$. If *Cn* is taken to be classical logic, the tree of Fig. 3 is not a derivation of $(a \wedge b, \{d, e\}, c)$ from Γ. The reason is that item 2 fails as follows. First, $\vdash f$ is a node of the derivation tree hence $f \in \{\alpha \mid \vdash \alpha \text{ is a node of } T\}$. Second, $(\top, \emptyset, \neg f)$ belongs to Γ hence $\top \supset \neg f \in \{\gamma_1 \supset \gamma_2 \mid (\top, \emptyset, \gamma_1 \supset \gamma_2) \in \Gamma\}$. Third, item 2 then becomes $\beta \notin Cn(\{f, \ldots, \top \supset \neg f\})$ that must be checked for β being d and e. However, as *Cn* is classical logic, $Cn(\{f, \ldots, \top \supset \neg f\})$ contains all formulas of the language, among them are d and e.

Example 11. Let $\Gamma = \{(a \wedge b, \{d, e\}, f), (f, \emptyset, d), (d, \{\neg c\}, c)\}$. The tree from Fig. 4 is not a derivation of $(a \wedge b, \{d, e, \neg c\}, c)$ from Γ because item 2 is violated. First, $\vdash d$ is a node of the derivation tree hence $d \in \{\alpha \mid \vdash \alpha \text{ is a node of } T\}$. As $\nvdash d$ is a leaf, $\beta \notin Cn(\{d, \ldots\})$ must be checked for β being d and failure is obvious.

5. A versatile approach

Again, it must be clear that the present work is *not* the definition of a new nonmonotonic logic and its proof theory. Instead, it is the definition of a framework expressive enough to capture an approach to the problem of overriding subsuming rules, and general enough to be instantiated by a number of logical formalisms. Importantly, the concept of a derivation is only a tool towards this aim which can be tailored to the proof theory of various logics.

For instance, and importantly, the above concept of a derivation does *not* match inference in Reiter's default logic. First, derivations have been defined independently of any extension concept and a derivation tree does not require all rules whose premises are satisfied to be applied and occur in the tree. In particular, it happens that derivations exist although there is no extension (cf Example 12).

$$\frac{\vdash b \quad \nvdash \neg d \quad \nvdash \neg e}{\vdash c}$$

Fig. 5. Tree from Example 12.

$$\frac{\vdash a \quad \nvdash b}{\vdash c}$$

Fig. 6. Tree from Example 13.

Example 12. Let $\Gamma = (\Delta, \Sigma)$ be a default theory with $\Delta = \{\frac{\top:a}{\neg a}, \frac{b:d,e}{c}\}$ and $\Sigma = \{b\}$. Let us represent Γ by the PEC rules $\Gamma' = \{(\top, \{\neg a\}, \neg a), (b, \{\neg d, \neg e\}, c), (\top, \emptyset, b)\}$. Γ has no extension because Δ contains the default $\frac{\top:a}{\neg a}$, yet there exists a derivation of the PEC rule $(\top, \{\neg d, \neg e\}, c)$ from Γ', as can be seen from the tree of Fig. 5.

Similarly, Example 13 shows that it may happen that a formula is in no extension although there exists a derivation for it.

Example 13. Let $\Gamma = (\Delta, \Sigma)$ be a default theory with $\Delta = \left\{\frac{\top:a}{b}, \frac{a:\neg b}{c}\right\}$ and $\Sigma = \{a\}$. Let us represent Γ by the set of PEC rules $\Gamma' = \{(\top, \{\neg a\}, b), (a, \{b\}, c), (\top, \emptyset, a)\}$. Γ has a single extension, i.e., $E = Cn(\{a, b\})$. Although the formula c is not in E, there exists a derivation of the PEC rule $(\top, \{b\}, c)$ from Γ' as shown by the tree of Fig. 6.

Still, the concept of a derivation is powerful enough to capture credulous reasoning as modeled by default logic, as the straightforward following property establishes it.

Property 1. *Let $\Gamma = (\Delta, \Sigma)$ be a default theory and $\Gamma^\star$ a set of PEC rules translating Γ. For every formula f belonging to an extension E of Γ, there exists a derivation $(\top, \{\epsilon_1, \ldots, \epsilon_n\}, f)$ from $\Gamma^\star$ where $\{\neg\epsilon_1, \ldots, \neg\epsilon_n\}$ is a subset of the justifications of the generating defaults of Γ with respect to E.*

Importantly, derivations are not meant to be optimal proofs: there is no endeavor as to avoid detours or to impose shortcuts. Lastly, a concept of consistency can be introduced into the PEC framework.

Definition 4 (*Consistency*)**.** Γ is consistent iff $\Gamma \not\mid\!\sim (\top, \emptyset, \bot)$.

As usual, a notion of consistency opens up a choice of negations. Whatever such a choice of a negation $\sim$ for PEC rules, it is likely to be such that both $\Gamma \mid\!\sim R$ and $\Gamma \mid\!\sim\ \sim\!R$ while $\Gamma \not\mid\!\sim R\ \&\ \sim\!R$ (where & stands for some conjunction of PEC rules, again whatever choice is made there) is possible. In purpose, we have thus left out any notion of inferential closure and similarly any subgrouping of consequences, e.g. in forms of extensions *à la* default logic.

6. *X*-derivation

We are now to extend the concept of a derivation by taking into account an additional hypothesis, which, in full generality, can be a PEC rule (with or without exceptions). This concept will be useful in our approach to preempting rules since we will have to consider what could be inferred provided that a given PEC rule that must not be subsumed is introduced inside Γ. This full-fledged account is called an X-derivation, the details of which are explained and more generally discussed after the formal definition below.

Definition 5 (*X-derivation*)**.** Let Γ be a set of PEC rules and X a PEC rule. An X-derivation of $\aleph = (\rho, \{\epsilon_1, \ldots, \epsilon_n\}, \varsigma)$ from Γ is a tree T whose nodes are signed standard Boolean formulas s.t.:

1. for every leaf of the form $\vdash \alpha$,
   - either $(\top, \emptyset, \alpha) \in \Gamma \cup \{X\}$,
   - or $(\alpha_1, \emptyset, \alpha_2) \in \Gamma \cup \{X\}$ and $\alpha = \alpha_1 \supset \alpha_2$,
   - or $\alpha = \rho$,
2. for every leaf of the form $\nvdash \beta$,
   $\beta \notin Cn(\{\gamma \text{ s.t. } (\top, \emptyset, \gamma) \in \Gamma \cup \{X\}\} \cup \{\gamma_1 \supset \gamma_2 \text{ s.t. } (\gamma_1, \emptyset, \gamma_2) \in \Gamma \cup \{X\}\}$
   $\cup\{\alpha \text{ s.t.} \vdash \alpha \text{ is a node of } T\})$,
3. if $\nvdash \beta$ is a node then it is a leaf,
4. every node $\vdash \alpha$ that is not a leaf has as parents,
   - either a tuple $(\vdash \alpha_1, \ldots, \vdash \alpha_k)$. In this case, $\alpha \in Cn(\{\alpha_1, \ldots, \alpha_k\})$
   - or a tuple $(\vdash \alpha_1, \nvdash \beta_1, \ldots, \nvdash \beta_m)$ where $m \geq 1$. In this case, $(\alpha_1, \{\beta_1, \ldots, \beta_m\}, \alpha) \in \Gamma \cup \{X\}$.
5. $\rho \in \{\alpha \text{ s.t.} \vdash \alpha \text{ is a leaf of } T\} \cup \{\top\}$, $\{\epsilon_1, \ldots, \epsilon_n\} = \{\beta \text{ s.t.} \nvdash \beta \text{ is a node of } T\}$ and $\vdash \varsigma$ is the root of T.

We note $\Gamma \mid\!\sim_X^{\{\epsilon_1,\ldots,\epsilon_n\}} \aleph$ and $\Gamma \mid\!\sim_X \aleph$ when $\{\epsilon_1, \ldots, \epsilon_n\}$ is empty.

$$\cfrac{\cfrac{\vdash a \wedge b \quad \nvdash d \quad \nvdash e}{\vdash f} \qquad \vdash f \supset c}{\vdash c}$$

Fig. 7. Tree from Examples 14, 15 and 17.

$$\cfrac{\cfrac{\cfrac{\vdash \top \supset a \wedge b}{\vdash a \wedge b} \quad \nvdash d \quad \nvdash e}{\vdash f} \qquad \vdash f \supset c}{\vdash c}$$

Fig. 8. Tree from Example 16.

$$\cfrac{\cfrac{\vdash a \wedge b \quad \nvdash d}{\vdash f} \qquad \nvdash e}{\vdash c}$$

Fig. 9. Tree from Example 18.

Actually, the PEC rule X can be regarded as supplementing the set of PEC rules Γ. Accordingly, if $X = (\top, \emptyset, \top)$, then an X-derivation of $\aleph$ from Γ happens to be a derivation of $\aleph$ from Γ (cf Example 14). The role of each of the three components of the derived rule $\aleph$ is detailed by item 5. Importantly, item 1 expresses that if a positive leaf (tautologies aside) is not some exception-free information encoded as a PEC rule from Γ then it is the premise of $\aleph$. Similarly to Definition 3, conditional reasoning can be conducted using exception-free information as an extra hypothesis, turning it into a positive leaf. However, the conditional piece can now be the X rule itself (more exactly, an equivalent form) when X represents a formula of classical logic for instance (cf Example 16). When X is a PEC rule $(\varrho, \{\xi_1, \ldots, \xi_h\}, \nu)$ that does have exceptions, if its premise ϱ stands as a positive leaf (i.e., $\vdash \varrho$) not issued from a rule in Γ (i.e., there exists no $(\kappa, \emptyset, \zeta)$ in Γ such that $\kappa \supset \zeta$ be ϱ), then ϱ turns out to be the premise of $\aleph$ (cf Example 18).

When X is used in the derivation and that the premise ϱ of X is not a leaf, then ϱ comes from a subproof in the tree (cf Example 19).

In all cases, when X is used in the derivation, its premise occurs (as an hypothesis or an intermediate conclusion) higher in the tree. Therefore, not only is X introduced as an extra hypothesis, but when it is mentioned in the tree, if its premise ϱ does not come from a subproof then $\vdash \varrho$ occurs as a leaf (and is regarded as established); hence ϱ enters the set of premises of $\aleph$ (where $\aleph$ is the PEC rule which is the conclusion of the derivation).

More generally, an X-derivation encompasses conditional reasoning in various forms because it involves consistency hypotheses, it can include an extra rule X, and assumes the premise of $\aleph$ (the PEC rule to be inferred).

Example 14. Let us return to Example 9, i.e., $\Gamma = \{(a \wedge b, \{d, e\}, f), (f, \emptyset, c)\}$. The tree of Fig. 7, reproduced from Example 9, is both a derivation and a $(\top, \emptyset, \top)$-derivation of $(a \wedge b, \{d, e\}, c)$ from Γ.

Example 15. Again, $\Gamma = \{(a \wedge b, \{d, e\}, f), (f, \emptyset, c)\}$ as in Example 9. The tree of Fig. 7 is a $(\top, \emptyset, a \wedge b)$-derivation of $(a \wedge b, \{d, e\}, c)$ from Γ, although in a rather vacuous way because the extra hypothesis $X = (\top, \emptyset, a \wedge b)$ is left unused. The tree of Fig. 7 is not a $(\top, \emptyset, a \wedge b)$-derivation of $(\top, \{d, e\}, c)$ from Γ. The reason is that item 1 in the definition of an X-derivation is not satisfied because $a \wedge b$ is not of the form $\alpha_1 \supset \alpha_2$ while $\rho = \top$. In contrast, the tree in the next example is a $(\top, \emptyset, a \wedge b)$-derivation of $(\top, \{d, e\}, c)$ from Γ.

Example 16. Let us still consider $\Gamma = \{(a \wedge b, \{d, e\}, f), (f, \emptyset, c)\}$. The tree of Fig. 8 is a $(\top, \emptyset, a \wedge b)$-derivation of $(\top, \{d, e\}, c)$ from Γ (informally meaning that assuming $a \wedge b$ allows us to conclude c, unless d or e be the case).

Example 17. Once more, $\Gamma = \{(a \wedge b, \{d, e\}, f), (f, \emptyset, c)\}$. The tree of Fig. 7 is a $(a \wedge b, \{d, e\}, f)$-derivation of $(a \wedge b, \{d, e\}, c)$ from Γ although in a rather vacuous way because $X = (a \wedge b, \{d, e\}, f)$ is in Γ. Indeed, the tree of Fig. 7 is also a $(a \wedge b, \{d, e\}, f)$-derivation of $(a \wedge b, \{d, e\}, c)$ from Γ' where Γ' is taken to be $\Gamma \setminus \{(a \wedge b, \{d, e\}, f)\}$.

Example 18. Let $\Gamma = \{(f, \{e\}, c)\}$. The tree of Fig. 9 is a $(a \wedge b, \{d\}, f)$-derivation of $(a \wedge b, \{d, e\}, c)$ from Γ. Please observe that the premise of X, namely $a \wedge b$, is not issued from Γ hence is also the premise of $\aleph$ (here, X is $(a \wedge b, \{d\}, f)$ and $\aleph$ is $(a \wedge b, \{d, e\}, c)$).

Example 19. Let $\Gamma = \{(a \wedge b, \{d, e\}, f)\}$. The tree of Fig. 10 is a $(f, \{g\}, c)$-derivation of $(a \wedge b, \{d, e, g\}, c)$ from Γ.

$$\dfrac{\dfrac{\vdash a \wedge b \quad \nvdash d \quad \nvdash e}{\vdash f} \qquad \nvdash g}{\vdash c}$$

Fig. 10. Tree from Example 19.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdash a \wedge b}{\vdash a} \quad \vdash a \supset g}{\vdash g} \quad \vdash h}{\vdash g \wedge h} \quad \vdash g \wedge h \supset c}{\vdash c}$$

Fig. 11. Tree from Example 20.

$$\dfrac{\dfrac{\vdash \top \supset g \quad \vdash \top \supset h}{\vdash g \wedge h} \quad \vdash g \wedge h \supset c}{\vdash c} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\vdash a \wedge b}{\vdash \top} \quad \vdash \top \supset g}{\vdash g} \quad \vdash \top \supset h}{\vdash g \wedge h} \quad \vdash g \wedge h \supset c}{\vdash c}$$

Fig. 12. Trees from Example 21.

7. PEC-implicants

PEC-implicants are intended to extend and adapt the usual concept of implicants modulo a set of formulas to the PEC formalism and with respect to the *X*-derivation mechanism. This concept will allow us to capture the idea of having one rule entailing another rule, as part of our problem dealing with subsuming rules. However, we believe that the PEC-implicants concepts can have various other applications, as they generalize the standard-logic concept of implicants to a more expressive generic nonmonotonic framework. Let us provide the definition before we explain the intuitions and motivations justifying it.

Definition 6 (*PEC-implicant*)**.** Let Γ be a set of unary PEC rules and $R = (\rho, \{\epsilon_1, \ldots, \epsilon_m\}, \varsigma)$ be a unary PEC rule. A unary PEC rule R' is a PEC-implicant of R modulo Γ iff there exists an R'-derivation $\mathcal{D}$ of (ρ, E^*, ς) from Γ s.t.

1. $\forall e' \in E^*, \exists e \in \{\epsilon_1, \ldots, \epsilon_m\}$ s.t. $e \in Cn(\{e'\})$,
2. $\forall e'' \in \{\epsilon_1, \ldots, \epsilon_m\} \setminus E^*, e'' \notin Cn\{\alpha \mid \vdash \alpha$ is a node of $\mathcal{D}\}$.

Definition 7 (*Strict PEC-implicant*)**.** Let Γ be a set of unary PEC rules. Let R and R' be two unary PEC rules. R' is a strict PEC-implicant of R modulo Γ iff R' is a PEC-implicant of R and R is not a PEC-implicant of R'.

To simplify matters, *Cn* stands for classical logic in all of the following examples.

First, let us note that in the monotonic fragment of the PEC formalism, i.e. when all rules exhibit empty sets of exceptions, neither item 1 nor item 2 does apply. Consequently, Definition 6 reduces to *$\mathcal{R}'$ is an implicant of $\mathcal{R}$ modulo Γ iff there exists an $\mathcal{R}'$-derivation of $\mathcal{R}$ from Γ* (noted $\Gamma \mid\!\sim_{\mathcal{R}'} \mathcal{R}$). Moreover, the derivation tree is a standard deduction one, possibly making use of $\mathcal{R}'$ and where the premise ρ of $\mathcal{R}$ can play the role of an additional hypothesis.

Example 20. Let $\Gamma = \{(g \wedge h, \emptyset, c), (\top, \emptyset, h)\}$ be a set of PEC rules. Let $\mathcal{R} = (a \wedge b, \emptyset, c)$ and $\mathcal{R}' = (a, \emptyset, g)$ be two PEC rules. $\mathcal{R}'$ is a PEC- implicant of $\mathcal{R}$ modulo Γ. Indeed, the tree of Fig. 11 is an $\mathcal{R}'$-derivation of $\mathcal{R}$ from Γ. As a leaf of the tree, the premises of $\mathcal{R}$ (namely $a \wedge b$) are thus assumed to be satisfied when the derivation is performed.

Example 21. Let $\Gamma = \{(g \wedge h, \emptyset, c), (\top, \emptyset, h)\}$ a set of PEC rules. Let $\mathcal{R} = (a \wedge b, \emptyset, c)$ and $\mathcal{R}' = (\top, \emptyset, g)$ be two PEC rules. The leftmost tree of Fig. 12 does not make $\mathcal{R}'$ to be a PEC-implicant of $\mathcal{R}$ modulo Γ. Indeed, this tree is not an $\mathcal{R}'$-derivation of $\mathcal{R}$ from Γ since the premise of $\mathcal{R}$ (i.e., $a \wedge b$) does not belong to the set of positive nodes of the tree. This violates the first part of item 7 in Definition 5. Note that $\mathcal{R}'$ is a (strict) PEC-implicant of $(\top, \emptyset, c)$ through the aforementioned tree. Now, $\mathcal{R}'$ is a PEC-implicant of $\mathcal{R}$ modulo Γ through the rightmost tree of Fig. 12. Indeed, this tree is an $\mathcal{R}'$-derivation of $\mathcal{R}$ from Γ. Note that, in this tree, we assume that the premise of $\mathcal{R}$ is satisfied, although it does not play a significant role here.

Example 22. Let $\Gamma = \{(a, \emptyset, c)\}$ be a set of PEC rules. Let $\mathcal{R} = (a \wedge b, \emptyset, c)$ and $\mathcal{R}' = (\top, \emptyset, x)$ be two PEC rules. $\mathcal{R}'$ is a PEC-implicant of $\mathcal{R}$ modulo Γ. Indeed, the tree from Fig. 13 is an $\mathcal{R}'$-derivation of $\mathcal{R}$ from Γ.Note that this tree is also a derivation of $\mathcal{R}$ from Γ.

$$\dfrac{\dfrac{\vdash a \wedge b}{\vdash a} \quad \vdash a \supset c}{\vdash c}$$

Fig. 13. Tree from Example 22.

$$\dfrac{\dfrac{\vdash a \wedge b}{\vdash a} \quad \nvdash \neg d}{\vdash c}$$

Fig. 14. Tree from Example 23.

$$\dfrac{\vdash a \quad \nvdash \neg d \wedge \neg f}{\vdash c}$$

Fig. 15. Tree from Example 24.

$$\dfrac{\dfrac{\vdash a \quad \nvdash \neg e \wedge \neg h}{\vdash g} \quad \nvdash \neg d \wedge \neg f}{\vdash c}$$

Fig. 16. Tree from Example 25.

Let us now consider the general case. When PEC rules with non empty sets of exceptions are considered, items 1 and 2 of Definition 6 apply. Intuitively, it is necessary to take into account the possible links between the exceptions of $\mathcal{R}$ and the exceptions of $\mathcal{R}'$. These links must be such that if $\mathcal{R}'$ does not apply due to one of its exceptions then $\mathcal{R}$ must not apply. On the contrary, if $\mathcal{R}$ does not apply due to one of its exceptions, it might be the case that $\mathcal{R}'$ applies. Thus, on the one hand, all exceptions of $\mathcal{R}'$ must be as strong as at least one exception of $\mathcal{R}$ (see $\neg d$, Example 23, $\neg d \wedge \neg f$, Examples 24 and 25). On the other hand, the exceptions occurring in the derivation tree used to show the implication and that are not implied by an exception of $\mathcal{R}'$ must also be as strong as at least one exception of $\mathcal{R}$ (see $\neg e \wedge \neg h$, Example 25). Thus, it is necessary to consider the possible links between the exceptions of $\mathcal{R}$ and the nodes in the derivation tree showing the implication. Our intuition leads us to require that the proof of the implication (through the derivation tree) does not entail an exception of $\mathcal{R}$ (see $\neg e$, Example 26 et e, Example 27).

Thus, showing that $\mathcal{R}'$ is a PEC-implicant of $\mathcal{R} = (\rho, \{\epsilon_1, \ldots, \epsilon_m\}, \varsigma)$ modulo Γ, amounts to showing that there exists an $\mathcal{R}'$-derivation of (ρ, E^*, ς) where E^* consists of the negative nodes of the tree (belonging or not to the set of exceptions of $\mathcal{R}'$), it must be the case that each element of E^* implies an exception of $\mathcal{R}$ (item 1) and it must be the case that no exception of $\mathcal{R}$ that is not in E^* is a logical consequence of the positive nodes in the tree. Note that if there exists a derivation of $\mathcal{R}$ from Γ (according to Definition 3) then for some PEC rules $\mathcal{R}'$, $\mathcal{R}'$ is a PEC-implicant of $\mathcal{R}$ modulo Γ, without $\mathcal{R}'$ playing any role in the derivation (see Example 29).

Example 23. Let Γ be an empty set of PEC rules. Let $\mathcal{R} = (a \wedge b, \{\neg d, \neg e\}, c)$ and $\mathcal{R}' = (a, \{\neg d\}, c)$ be two PEC rules. $\mathcal{R}'$ is a PEC-implicant of $\mathcal{R}$ modulo Γ through the tree of Fig. 14. Indeed, this tree is an $\mathcal{R}'$-derivation of $(a \wedge b, \{\neg d\}, c)$ from Γ. In accordance with item 1, $\neg d$ (exception of $\mathcal{R}'$) belongs to the set of exceptions of $\mathcal{R}$ and in accordance with item 2, $\neg e$ cannot be deduced from $\{a, a \wedge b, c\}$.

Example 24. Let Γ be an empty set of PEC rules. Let $\mathcal{R} = (a, \{\neg d, \neg e\}, c)$ and $\mathcal{R}' = (a, \{\neg d \wedge \neg f\}, c)$ be two PEC rules. $\mathcal{R}'$ is a PEC-implicant of $\mathcal{R}$ modulo Γ, as shown by the tree of Fig. 15. Indeed, this tree is an $\mathcal{R}'$-derivation of $(a, \{\neg d \wedge \neg f\}, c)$ from Γ. In accordance with item 1, $\neg d \wedge \neg f$ (exception of $\mathcal{R}'$) implies $\neg d$ (an exception of $\mathcal{R}$) and in accordance with item 2, neither $\neg d$ nor $\neg e$ (the members of $\{\epsilon_1, \ldots, \epsilon_m\}$) are implied by $\{a, c\}$ (the formulas attached to the positive nodes of the tree).

Example 25. Let $\Gamma = \{(a, \{\neg e \wedge \neg h\}, g)\}$ be a set of PEC rules. Let $\mathcal{R} = (a, \{\neg d, \neg e\}, c)$ and $\mathcal{R}' = (g, \{\neg d \wedge \neg f\}, c)$ be two PEC rules. $\mathcal{R}'$ is a PEC-implicant of $\mathcal{R}$ modulo Γ, as illustrated by the tree of Fig. 16. Indeed, this tree is an $\mathcal{R}'$-derivation of $(a, \{\neg d \wedge \neg f, \neg e \wedge \neg h\}, c)$ from Γ. In accordance with item 1, $\neg d \wedge \neg f$ (which is in E^* and is an exception of $\mathcal{R}'$) implies $\neg d$ (an exception of $\mathcal{R}$) and $\neg e \wedge \neg h$ (which is E^* and is an exception of the only PEC rule of Γ) implies $\neg e$ (an exception of $\mathcal{R}$). In accordance with item 2, neither $\neg d$ nor $\neg e$ (the members of $\{\epsilon_1, \ldots, \epsilon_m\}$) are implied by $\{a, c, g\}$ (formulas corresponding to the positive nodes of the tree).

Example 26. Let $\Gamma = \{(a, \emptyset, \neg e), (\neg e, \emptyset, f)\}$ be a set of PEC rules. Let $\mathcal{R} = (a \wedge b, \{\neg d, \neg e\}, c)$ and $\mathcal{R}' = (f, \{\neg d\}, c)$ be two PEC rules. $\mathcal{R}'$ is not a PEC-implicant of $\mathcal{R}$ modulo Γ through the tree of Fig. 17. Indeed, although this tree is an $\mathcal{R}'$-derivation of $(a \wedge b, \{\neg d\}, c)$ from Γ, the item 2 from Definition 6 is not respected: $\vdash \neg e$ is a node of the tree although $\neg e$ is an exception of $\mathcal{R}$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\vdash a \wedge b}{\vdash a} \quad \vdash a \supset \neg e}{\vdash \neg e} \quad \vdash \neg e \supset f}{\vdash f} \quad \not\vdash \neg d}{\vdash c}$$

Fig. 17. Tree from Example 26.

$$\dfrac{\dfrac{\dfrac{\vdash a \wedge b}{\vdash a} \quad \not\vdash e}{\vdash f} \quad \not\vdash \neg d}{\vdash c}$$

Fig. 18. Tree from Example 27.

$$\dfrac{\vdash a \supset b \quad \vdash a}{\vdash b}$$

Fig. 19. Tree from Example 28.

$$\dfrac{\dfrac{\vdash a \wedge b}{\vdash a} \quad \not\vdash d \quad \not\vdash e}{\vdash c}$$

Fig. 20. Tree from Example 29.

Example 27. Let $\Gamma = \{(a, \{e\}, f)\}$ be a set of PEC rules. Let $\mathcal{R} = (a \wedge b, \{\neg d, \neg e\}, c)$ and $\mathcal{R}' = (f, \{\neg d\}, c)$ be two PEC rules. $\mathcal{R}'$ is not a PEC-implicant of $\mathcal{R}$ modulo Γ through the tree of Fig. 18. Although the latter one is an $\mathcal{R}'$-derivation of $(a \wedge b, \{\neg d, e\}, c)$ from Γ, item 1 from Definition 6 is violated: e is a formula of E^* from which no formula from $\{\neg d, \neg e\}$ can be inferred.

Example 28. Let Γ be an empty set of PEC rules. Let $\mathcal{R} = (a, \{\neg b\}, b)$ and $\mathcal{R}' = (\top, \emptyset, a \supset b)$ be two PEC rules. $\mathcal{R}'$ is a PEC-implicant of $\mathcal{R}$ modulo Γ through the tree from Fig. 19. Indeed, this tree is an $\mathcal{R}'$-derivation of $(a, \emptyset, b)$ from Γ. Item 1 is satisfied since E^* is empty. Moreover, in accordance with item 2, $\neg b$ cannot be deduced from $\{a, b, a \supset b\}$ (i.e., the formulas attached to the positive nodes of the tree).

Example 29. Let $\Gamma = \{(a, \{d, e\}, c)\}$ be a set of PEC rules. Let $\mathcal{R} = (a \wedge b, \{d, e\}, c)$ and $\mathcal{R}' = (x, \{z\}, y)$ be two PEC rules. $\mathcal{R}'$ is a PEC-implicant of $\mathcal{R}$ modulo Γ. Indeed, the tree from Fig. 20 is an $\mathcal{R}'$-derivation of $\mathcal{R}$ from Γ. Let us stress that here $\mathcal{R}'$ does not play any role in the derivation: This tree is also a derivation of $\mathcal{R}$ from Γ.

Fairly weak requirements about *Cn* are enough to show that being a PEC-implicant defines a pre-order. Of special interest then is the case that two PEC rules are PEC-implicants of each other: They surely are equivalent in a strong sense closely related to *Cn*-equivalence of exceptions. It is straightforward to obtain such a result, as follows.

Property 2. *Given two unary PEC rules $\mathcal{R} = (\rho, \{\epsilon_1, \ldots, \epsilon_m\}, \varsigma)$ and $\mathcal{R}' = (\rho', \{\epsilon'_1, \ldots, \epsilon'_n\}, \varsigma')$, if $\mathcal{R}$ is a PEC-implicant of $\mathcal{R}'$ modulo Γ, and $\mathcal{R}'$ is a PEC-implicant of $\mathcal{R}$ modulo Γ where $\Gamma = \emptyset$ then:*

1. $\rho \Vdash\!\!\vdash \rho'$ *and* $\rho' \Vdash\!\!\vdash \rho$,
2. $\forall \epsilon_i \in \{\epsilon_1, \ldots, \epsilon_m\}, \exists \epsilon_j, \epsilon'_k$ *where* $\epsilon_j \in \{\epsilon_1, \ldots, \epsilon_m\}$ *and* $\epsilon'_k \in \{\epsilon'_1, \ldots, \epsilon'_n\}$, *s.t.* $\epsilon_j \Vdash\!\!\vdash \epsilon_i$ *and* $\epsilon_j \Vdash\!\!\vdash \epsilon'_k$ *and* $\epsilon'_k \Vdash\!\!\vdash \epsilon_j$.

The idea underlying the previous result is to extend to PEC rules the idea that when two objects are symmetrically related with each other through a given relation, then they must be equivalent in some sense (other than the trivial equivalence obtained from the binary relation under consideration). In particular, item 2 means that exceptions in $\mathcal{R}$ and $\mathcal{R}'$ are the same, up to logical equivalence (by subsumption, there can be more exceptions in $\mathcal{R}$ or in $\mathcal{R}'$, though).

8. Essential implicants and prime implicants

From now on, we write implicant as a shorthand for PEC-implicant.

We have seen that some implicants only play a superfluous role in the X-derivation used to show the implication. According to our goal to insert a PEC rule while ensuring that rules subsuming it are rejected, we should not discard those implicants. To this end, we define a concept of essential implicant that does not select them.

We assume that an operator $\setminus$ is available in the PEC framework. Intuitively, $\setminus$ is a syntactical contraction operator between a set of PEC rules and a PEC rule: $\Gamma \setminus \mathcal{R}$ contracts Γ from $\mathcal{R}$ (and from any equivalent PEC rule to $\mathcal{R}$, in accordance with the equivalence concept introduced at the end of the previous Section).

$$\dfrac{\dfrac{\vdash a \wedge b}{\vdash a} \qquad \nvdash \neg d \wedge \neg f}{\vdash c} \qquad \dfrac{\dfrac{\vdash a \wedge b}{\vdash a} \qquad \vdash a \supset c}{\vdash c} \qquad \dfrac{\vdash a \quad \vdash a \supset c}{\vdash c}$$

Fig. 21. Trees from Example 30.

Definition 8 (*Essential implicant*)**.** Let Γ be a set of PEC rules, $\mathcal{R}$ and $\mathcal{R}'$ be PEC rules. $\mathcal{R}'$ is an essential implicant of $\mathcal{R}$ modulo Γ iff there exists $\Gamma' \subseteq \Gamma \setminus \{\mathcal{R}\}$ s.t. :

1. $\mathcal{R}'$ is an implicant of $\mathcal{R}$ modulo Γ',
2. there does not exist any derivation of an implicant of $\mathcal{R}$ modulo $\Gamma' \setminus \{\mathcal{R}'\}$.

Definition 9 (*Strict essential implicant*)**.** Let Γ be a set of PEC rules. Let $\mathcal{R}$ and $\mathcal{R}'$ be two PEC rules. $\mathcal{R}'$ is a strict essential implicant of $\mathcal{R}$ modulo Γ iff $\mathcal{R}'$ is an essential implicant of $\mathcal{R}$ modulo Γ and $\mathcal{R}$ is not an essential implicant of $\mathcal{R}'$ modulo Γ.

Then, for all strict essential implicants $\mathcal{R}'$ of $\mathcal{R}$ modulo Γ, we say that $\mathcal{R}'$ is a prime implicant of $\mathcal{R}$ modulo Γ if there does not exist any rule $\mathcal{R}''$ that is an essential implicant of $\mathcal{R}$ modulo Γ, s.t. $\mathcal{R}'$ would be one of its essential implicant modulo Γ.

Definition 10 (*Prime implicant*)**.** Let Γ be a set of PEC rules, $\mathcal{R}$ and $\mathcal{R}'$ be two PEC rules. $\mathcal{R}'$ is a prime implicant of $\mathcal{R}$ modulo Γ iff $\mathcal{R}'$ is a strict essential implicant of $\mathcal{R}$ modulo Γ and there does not exist any PEC rule $\mathcal{R}''$ s.t.:

1. $\mathcal{R}''$ is a strict essential implicant of $\mathcal{R}$ modulo Γ,
2. $\mathcal{R}'$ is a strict essential implicant of $\mathcal{R}''$ modulo Γ.

Let us provide the reader with some intuitions about the above definitions. First, consider the specific case where Γ is empty. The definition for essential implicant gets simpler: item 2 does not apply and only one Γ' exists (and it is empty). Intuitively, when Γ is empty, any implicant $\mathcal{R}'$ of $\mathcal{R}$ modulo Γ is an essential implicant of $\mathcal{R}$ modulo Γ (see Example 30). Note that in this case, a prime implicant of $\mathcal{R}$ modulo Γ is a strict implicant $\mathcal{R}'$ of $\mathcal{R}$ modulo Γ s.t. there does not exist any other strict implicant $\mathcal{R}''$ of $\mathcal{R}$ of which $\mathcal{R}'$ is a strict implicant modulo Γ. Intuitively, when Γ is not empty, in order for an implicant of $\mathcal{R}'$ of $\mathcal{R}$ modulo Γ to be an essential one, it is also necessary, leaving $\mathcal{R}'$ apart, that no way to derive $\mathcal{R}$ (or one of its implicants) remains.

Example 30. Let Γ be an empty set of PEC rules and $\mathcal{R} = (a \wedge b, \{\neg d, \neg e\}, c)$ be a PEC rule. First, consider the essential implicants of $\mathcal{R}$ modulo Γ. Note that $\Gamma' = \Gamma \setminus \{\mathcal{R}\} = \Gamma = \emptyset$. The PEC rule $\mathcal{R}' = (a, \{\neg d \wedge \neg f\}, c)$ is an essential implicant of $\mathcal{R}$ modulo Γ. On the one hand, $\mathcal{R}'$ is an implicant of $\mathcal{R}$ modulo Γ'. Indeed, as the leftmost tree of Fig. 21 shows, there exists an $\mathcal{R}'$-derivation of $(a \wedge b, \{\neg d \wedge \neg f\}, c)$ from Γ'. On the other hand, there cannot exist any derivation of $\mathcal{R}$ or of one of its implicants from $\Gamma' \setminus \{\mathcal{R}'\}$ since $\Gamma = \emptyset$. Similarly, the PEC rule $\mathcal{R}'' = (a, \emptyset, c)$ is an essential implicant of $\mathcal{R}$ modulo Γ. On the one hand, $\mathcal{R}''$ is an implicant of $\mathcal{R}$ modulo Γ'. Indeed, as the middle tree of Fig. 21 shows, there exists an $\mathcal{R}''$-derivation of $(a \wedge b, \emptyset, c)$ from Γ'. On the other hand, there cannot exist any derivation of $\mathcal{R}$ or of any of its implicants, from $\Gamma' \setminus \{\mathcal{R}''\}$. Now, consider the essential implicants of $\mathcal{R}'$ modulo Γ. Note that $\Gamma' = \Gamma \setminus \{\mathcal{R}'\} = \Gamma = \emptyset$. First, it is easy to see that $\mathcal{R}$ is not an essential implicant of $\mathcal{R}'$ modulo Γ. $\mathcal{R}'$ is thus a strict essential implicant of $\mathcal{R}$ modulo Γ. Moreover, $\mathcal{R}''$ is an essential implicant of $\mathcal{R}'$ modulo Γ. On the one hand, $\mathcal{R}''$ is an implicant of $\mathcal{R}'$ modulo Γ'. Indeed, there exists an $\mathcal{R}''$-derivation of $(a, \emptyset, c)$ from Γ', as the rightmost tree of Fig. 21 shows. On the other hand, there cannot exist any derivation of $\mathcal{R}'$ or of any of its implicants, modulo $\Gamma' \setminus \{\mathcal{R}''\}$ since $\Gamma' \setminus \{\mathcal{R}''\}$ is empty. Consider now the essential implicants of $\mathcal{R}''$ modulo Γ. Note that $\Gamma' = \Gamma \setminus \{\mathcal{R}''\} = \Gamma = \emptyset$. It is easy to see that, at the same time, $\mathcal{R}'$ and $\mathcal{R}$ are not essential implicants of $\mathcal{R}''$ modulo Γ'. Thus, $\mathcal{R}''$ is both a strict essential implicant of $\mathcal{R}$ and of $\mathcal{R}'$ modulo Γ. Consequently, $\mathcal{R}''$ is not a prime implicant of $\mathcal{R}$ modulo Γ. Also, note that although $\mathcal{R}'$ is a strict essential implicant of $\mathcal{R}$ modulo Γ, $\mathcal{R}'$ is not necessarily a prime implicant of $\mathcal{R}$ modulo Γ. Indeed, other strict essential implicants of $\mathcal{R}$ modulo Γ can exist, of which $\mathcal{R}'$ is a strict essential implicant modulo Γ (e.g., the PEC rule $\mathcal{R}''' = (a \wedge b, \{\neg d \wedge \neg f\}, c)$ is a strict essential implicant of $\mathcal{R}$ modulo Γ and $\mathcal{R}'$ is a strict essential implicant of $\mathcal{R}'''$ modulo Γ).

Example 31. Let $\Gamma = \{(a, \{\neg d, \neg e\}, c)\}$ be a set of PEC rules and $\mathcal{R} = (a \wedge b, \{\neg d, \neg e\}, c)$ a PEC rule. First, let us consider the essential implicants of $\mathcal{R}$ modulo Γ. Note that here $\Gamma \setminus \{\mathcal{R}\} = \Gamma$. Thus, there can only exist two different Γ', namely:

- $\Gamma'_1 = \Gamma$,
- $\Gamma'_2 = \emptyset$.

The PEC rule $\mathcal{R}' = (x, \{\neg z\}, y)$ is not an essential implicant of $\mathcal{R}$ modulo Γ. First, consider Γ'_1. On the one hand, $\mathcal{R}'$ is an implicant of $\mathcal{R}$ modulo Γ'_1. Indeed, as shown in the tree of Fig. 22, there exists an $\mathcal{R}'$-derivation of $\mathcal{R}$ modulo Γ'_1. But on the other hand, as the same tree shows it, there exists a derivation of $\mathcal{R}$ modulo $\Gamma'_1 \setminus \{\mathcal{R}'\}$ (note that $\Gamma'_1 \setminus \{\mathcal{R}'\} = \Gamma$). Now, consider Γ'_2. Clearly, there does not exist any $\mathcal{R}'$-derivation of $\mathcal{R}$ modulo Γ'_2. Thus, $\mathcal{R}'$ is not an implicant of $\mathcal{R}$ modulo Γ'_2.

$$\dfrac{\dfrac{\vdash a \wedge b}{\vdash a} \qquad \nvdash \neg d \quad \nvdash \neg e}{\vdash c}$$

Fig. 22. Tree from Example 31.

$$\dfrac{\dfrac{\dfrac{\dfrac{\vdash a \wedge b}{\vdash a} \qquad \nvdash \neg d \wedge \neg f}{\vdash c \wedge x}}{\vdash x} \qquad \nvdash \neg d \wedge \neg f}{\vdash c} \qquad\qquad \dfrac{\dfrac{\dfrac{\vdash a \wedge b}{\vdash a} \qquad \nvdash \neg d \wedge \neg f}{\vdash c \wedge x}}{\vdash c}$$

Fig. 23. Trees from Example 32.

Accordingly, $\mathcal{R}'$ is not a strict essential implicant of $\mathcal{R}$ modulo Γ and it thus not a prime implicant. On the contrary, the PEC rule $\mathcal{R}'' = (a, \{\neg d, \neg e\}, c)$ is an essential implicant of $\mathcal{R}$ modulo Γ. On the one hand, $\mathcal{R}''$ is an implicant of $\mathcal{R}$ modulo Γ'_1. Indeed, the tree from Fig. 22 is an $\mathcal{R}''$-derivation of $\mathcal{R}$ modulo Γ'_1. On the other hand, there does not exist any derivation of $\mathcal{R}$ or of one of its implicants modulo $\Gamma'_1 \setminus \{\mathcal{R}''\}$. Indeed in this case $\Gamma'_1 \setminus \{\mathcal{R}''\} = \emptyset$. Consider now the essential implicants $\mathcal{R}''$ modulo Γ. Note that in this case $\Gamma \setminus \{\mathcal{R}''\} = \emptyset$. Thus, there exists only one Γ' and it is empty. It is easy to see that $\mathcal{R}$ is not an essential implicant of $\mathcal{R}''$ modulo Γ. $\mathcal{R}''$ is thus a strict essential implicant of $\mathcal{R}$ modulo Γ. Thus, amongst $\{\mathcal{R}', \mathcal{R}''\}$, only $\mathcal{R}''$ and its equivalent forms can be candidates for being prime implicants of $\mathcal{R}$ modulo Γ.

Example 32. Let $\Gamma = \{(a, \{\neg d \wedge \neg f\}, c \wedge x)\}$ be a set of PEC rules and $\mathcal{R} = (a \wedge b, \{\neg d, \neg e\}, c)$ one PEC rule. First, consider the essential implicants of $\mathcal{R}$ modulo Γ. Note that here $\Gamma \setminus \{\mathcal{R}\} = \Gamma$. Thus, only two different Γ' exist, namely:

- $\Gamma'_1 = \Gamma$,
- $\Gamma'_2 = \emptyset$.

The PEC rule $\mathcal{R}' = (x, \{\neg d \wedge \neg f\}, c)$ is not an essential implicant of $\mathcal{R}$ modulo Γ. First, consider Γ'_1. On the one hand $\mathcal{R}'$ is an implicant of $\mathcal{R}$ modulo Γ'_1. Indeed, as the leftmost tree of Fig. 23 shows, there exists an $\mathcal{R}'$-derivation of $(a \wedge b, \{\neg d \wedge \neg f\}, c)$ modulo Γ'_1. On the other hand, as the rightmost tree of Fig. 23 shows, there exists a derivation of $(a \wedge b, \{\neg d \wedge \neg f\}, c)$, an implicant of $\mathcal{R}$, modulo $\Gamma'_1 \setminus \{\mathcal{R}'\}$ (note that $\Gamma'_1 \setminus \{\mathcal{R}'\} = \Gamma$). Now, consider Γ'_2. It is clear that there does not exist any $\mathcal{R}'$-derivation of $\mathcal{R}$ modulo Γ'_2. Thus, $\mathcal{R}'$ is not an implicant of $\mathcal{R}$ modulo Γ'_2. Consequently, $\mathcal{R}'$ is not a strict essential implicant of $\mathcal{R}$ modulo Γ hence is not a prime implicant. On the contrary, the PEC rule $\mathcal{R}'' = (a, \{\neg d \wedge \neg f\}, c \wedge x)$ is an essential implicant of $\mathcal{R}$ modulo Γ. On the one hand, $\mathcal{R}''$ is an implicant of $\mathcal{R}$ modulo Γ'_1. Indeed, the rightmost tree of Fig. 23 is an $\mathcal{R}''$-derivation of $(a \wedge b, \{\neg d \wedge \neg f\}, c)$ modulo Γ'_1. On the other hand, there does not exist any derivation of $\mathcal{R}$ or of any of its implicants modulo $\Gamma'_1 \setminus \{\mathcal{R}''\}$. Indeed, in this case, $\Gamma'_1 \setminus \{\mathcal{R}''\} = \emptyset$. Now, consider the essential implicants of $\mathcal{R}''$ modulo Γ. Note that here $\Gamma \setminus \{\mathcal{R}''\} = \emptyset$. Thus, there exists only one Γ' and it is empty. It is easy to see that $\mathcal{R}$ is not an essential implicant of $\mathcal{R}''$ modulo Γ. $\mathcal{R}''$ is thus a strict essential implicant of $\mathcal{R}$ modulo Γ. Thus, amongst $\{\mathcal{R}', \mathcal{R}''\}$, only $\mathcal{R}''$ and its equivalent forms can be candidates for being prime implicants of $\mathcal{R}$ modulo Γ.

Example 33. Let $\Gamma = \{(a, \{\neg d \wedge \neg f\}, c), (a \vee h, \emptyset, c)\}$ be a set of PEC rules and $\mathcal{R} = (a \wedge b, \{\neg d, \neg e\}, c)$ a PEC rule. First, consider the essential implicants of $\mathcal{R}$ modulo Γ. Note that here $\Gamma \setminus \{\mathcal{R}\} = \Gamma$. Thus, there exists four different Γ', namely:

- $\Gamma'_1 = \Gamma$,
- $\Gamma'_2 = \{(a, \{\neg d \wedge \neg f\}, c)\}$,
- $\Gamma'_3 = \{(a \vee h, \emptyset, c)\}$,
- $\Gamma'_4 = \emptyset$.

The PEC rule $\mathcal{R}' = (a, \{\neg d \wedge \neg f\}, c)$ is an essential implicant of $\mathcal{R}$ modulo Γ. On the one hand, $\mathcal{R}'$ is an implicant of $\mathcal{R}$ modulo Γ'_2 (or Γ'_4). Indeed, as the leftmost tree of Fig. 24 shows it, there exists an $\mathcal{R}'$-derivation of $(a \wedge b, \{\neg d \wedge \neg f\}, c)$ modulo Γ'_2 (or Γ'_4). On the other hand, there does not exist any derivation of $\mathcal{R}$ or of one of its implicants modulo $\Gamma'_2 \setminus \{\mathcal{R}'\}$ since $\Gamma'_2 \setminus \{\mathcal{R}'\} = \Gamma'_4 = \emptyset$. Similarly, the PEC rule $\mathcal{R}'' = (a \vee h, \emptyset, c)$ is an essential implicant of $\mathcal{R}$ modulo Γ. On the one hand $\mathcal{R}''$ is an implicant of $\mathcal{R}$ modulo Γ'_3 (or Γ'_4). Indeed, as the middle tree of Fig. 24 shows it, there exists an $\mathcal{R}''$-derivation of $(a \wedge b, \emptyset, c)$ modulo Γ'_3 (or Γ'_4). On the other hand, there does not exist any derivation of $\mathcal{R}$ or of one of its implicants modulo $\Gamma'_3 \setminus \{\mathcal{R}''\}$ since $\Gamma'_3 \setminus \{\mathcal{R}''\} = \Gamma'_4 = \emptyset$. Now consider the essential implicants of $\mathcal{R}'$ modulo Γ. Note that here $\Gamma \setminus \{\mathcal{R}'\} = \{(a \vee h, \emptyset, c)\}$. Thus, there exists only two different Γ', namely:

- $\Gamma'_1 = \{(a \vee h, \emptyset, c)\}$,
- $\Gamma'_2 = \emptyset$.

First, it is easy to see that $\mathcal{R}$ is not an essential implicant of $\mathcal{R}'$ modulo Γ. $\mathcal{R}'$ is thus a strict essential implicant of $\mathcal{R}$ modulo Γ. Now, $\mathcal{R}''$ is an essential implicant of $\mathcal{R}'$ modulo Γ. On the one hand, $\mathcal{R}''$ is an implicant of $\mathcal{R}'$ modulo Γ'_1. Indeed, as the

$$\dfrac{\dfrac{\vdash a \wedge b}{\vdash a} \quad \nvdash \neg d \wedge \neg f}{\vdash c} \qquad \dfrac{\dfrac{\dfrac{\vdash a \wedge b}{\vdash a}}{\vdash a \vee h} \quad \vdash (a \vee h) \supset c}{\vdash c} \qquad \dfrac{\dfrac{\vdash a}{\vdash a \vee h} \quad \vdash (a \vee h) \supset c}{\vdash c}$$

Fig. 24. Trees from Example 33.

rightmost tree of Fig. 24 shows, there exists an $\mathcal{R}''$-derivation of $(a, \emptyset, c)$ modulo Γ'_1. On the other hand, there does not exist any derivation of $\mathcal{R}'$ or of one of its implicants modulo $\Gamma'_1 \setminus \{\mathcal{R}''\}$ since $\Gamma'_1 \setminus \{\mathcal{R}''\} = \emptyset$. Now consider the essential implicants of $\mathcal{R}''$ modulo Γ. Note that here $\Gamma \setminus \{\mathcal{R}''\} = \{(a, \{\neg d \wedge \neg f\}, c)\}$. Thus, there exist only two different Γ', namely:

- $\Gamma'_1 = \{(a, \{\neg d \wedge \neg f\}, c)\}$,
- $\Gamma'_2 = \emptyset$.

Again, it is easy to see that $\mathcal{R}$ is not an essential implicant of $\mathcal{R}''$ modulo Γ. $\mathcal{R}''$ is thus a strict essential implicant of $\mathcal{R}$ modulo Γ. However, $\mathcal{R}'$ is not an essential implicant of $\mathcal{R}''$ modulo Γ. To show it, it is useful to consider the subsets Γ' of $\Gamma \setminus \{\mathcal{R}''\}$:

- Consider $\Gamma'_1 = \{(a, \{\neg d \wedge \neg f\}, c)\}$. Clearly, $\mathcal{R}'$ is not an implicant of $\mathcal{R}''$ modulo Γ'_1.
- Consider $\Gamma'_2 = \emptyset$. Clearly, $\mathcal{R}'$ is not an implicant of $\mathcal{R}''$ modulo Γ'_2.

Thus, $\mathcal{R}''$ is a strict essential implicant of $\mathcal{R}'$ modulo Γ. Consequently, $\mathcal{R}''$ cannot be a prime implicant of $\mathcal{R}$ modulo Γ

9. Overriding subsuming rules

We are now ready to introduce our approach to override subsuming rules. To override the subsuming rules of a PEC rule *R* and make *R* to preempt, it is presumably not sufficient to "withdraw" all prime implicants of *R* and insert *R*. Indeed, as shown in the next example, there may remain in the resulting Γ some information of a self-conflicting change, e.g. so that whenever *R* is derivable, one of its strict PEC-implicants is also derivable.

Example 34. Let $\Gamma = \{(\top, \emptyset, c \supset a \vee b)\}$ and $\mathcal{R} = (\top, \emptyset, a \vee b \vee c)$. Assume that we want to augment Γ with $\mathcal{R}$ in such a way that $\mathcal{R}$ is not subsumed. Clearly, there is no prime implicant of $\mathcal{R}$ in Γ. However, introducing $\mathcal{R}$ in Γ will allow $a \vee b$ to be deduced, which is a strict implicant of $a \vee b \vee c$.

Accordingly, the process will thus be a little more elaborate. The intuition behind the approach is best explained in the Boolean framework: assume that a formula *g* is to be introduced inside Γ so that *g* is not subsumed. In the case *g* is consistent with Γ, it is necessary to retract $g \supset f$ for every prime implicant *f* of *g*. Intuitively, when *g* is then inserted inside Γ, no way to infer *f* remains available. Such an idea has been explored for the Boolean framework in [3].

Let us extend this idea to the full PEC-framework and *X*-derivation mechanism. In the sequel, we assume an operator $\setminus$ to be available in the PEC framework with the following features. Intuitively, $\setminus$ is a kind of contraction operator which applies to a set of PEC rules and to a pair of PEC rules: $\Gamma \setminus (R, R')$ is intended to contract Γ of R' in the presence of *R*. Formally, it is expected to enjoy the following properties.

1. $\Gamma \setminus (\mathcal{R}, \mathcal{R}') \not\mid\!\sim^{\mathcal{E}}_{\mathcal{R}} \mathcal{R}'$,
2. $\Gamma \setminus (\mathcal{R}, \mathcal{R}') \mid\!\sim^{\mathcal{E}} \mathcal{R}''$ implies $\Gamma \mid\!\sim^{\mathcal{E}} \mathcal{R}''$,
3. $\Gamma \setminus (\mathcal{R}, \mathcal{R}') = Cn(\Gamma \setminus (\mathcal{R}, \mathcal{R}'))$,
4. For $\mathcal{R} = (\rho, \{\epsilon_1, \ldots, \epsilon_n\}, \varsigma)$ where $n \geq 1$, if $\epsilon_i \notin Cn(\{\rho, \varsigma\})$ for $i = 1..n$ then $\Gamma \setminus (\mathcal{R}, \mathcal{R}') \not\mid\!\sim (\rho \wedge \varsigma, \emptyset, \epsilon_i)$ for $i = 1..n$,

where $\mathcal{R}, \mathcal{R}', \mathcal{R}''$ are PEC rules and Γ is a set of PEC rules that does not need to be consistent. The last property is not natural for a pure contraction operator. However, it proves convenient in allowing us to state the following natural definition.

Definition 11. Let $\mathcal{R}'$ be a prime PEC-implicant of $\mathcal{R}$ modulo Γ.
$\Gamma \oplus_{\rangle \mathcal{R}' \langle} \mathcal{R} =_{def} \Gamma \setminus (\mathcal{R}, \mathcal{R}') \cup \{\mathcal{R}\}$.

Theorem 1. *Let Γ be a set of PEC rules. Let $\mathcal{R} = (\rho, \{\epsilon_1, \ldots, \epsilon_n\}, \varsigma)$ a PEC rule s.t. $n \geq 1$ and s.t. $\epsilon_i \notin Cn(\{\rho, \varsigma\})$ for $i = 1..n$. Let $\mathcal{R}'$ be a prime implicant of $\mathcal{R}$ modulo Γ.*

- *$\Gamma \oplus_{\rangle \mathcal{R}' \langle} \mathcal{R}$ is consistent,*
- *$\Gamma \oplus_{\rangle \mathcal{R}' \langle} \mathcal{R} \mid\!\sim^{\epsilon} \mathcal{R}$,*
- *$\Gamma \oplus_{\rangle \mathcal{R}' \langle} \mathcal{R} \not\mid\!\sim^{\epsilon} \mathcal{R}'$.*

Proof. First let us show that $\Gamma \oplus_{\rangle\mathcal{R}'\langle} \mathcal{R} \mid\!\sim^{\epsilon} \mathcal{R}$. According to Definition 11, $\mathcal{R} \in \Gamma \oplus_{\rangle\mathcal{R}'\langle} \mathcal{R}$. Thus, we can build a tree whose root is $\vdash \varsigma$ and whose parents are $\vdash \rho$ and $\nvdash \epsilon_1, \ldots, \nvdash \epsilon_n$. This tree clearly obeys all conditions of Definition 3: only item 2 requires more analysis. We need to prove that $\epsilon_i \notin Cn(\{\gamma \text{ s.t. } (\top, \emptyset, \gamma) \in \Gamma \setminus (\mathcal{R}, \mathcal{R}')\} \cup \{\gamma_1 \supset \gamma_2 \text{ s.t. } (\gamma_1, \emptyset, \gamma_2) \in \Gamma \setminus (\mathcal{R}, \mathcal{R}')\} \cup \{\rho, \varsigma\})$.This is equivalent to $\epsilon_i \notin Cn(\{\gamma_1 \supset \gamma_2 \text{ t.q. } (\gamma_1, \emptyset, \gamma_2) \in \Gamma \setminus (\mathcal{R}, \mathcal{R}')\} \cup \{\rho, \varsigma\})$ and to $\{\gamma_1 \supset \gamma_2 \text{ t.q. } (\gamma_1, \emptyset, \gamma_2) \in \Gamma \setminus (\mathcal{R}, \mathcal{R}')\} \not\models \rho \wedge \varsigma \supset \epsilon_i$. The latter condition is satisfied through the constraint that $\setminus(\mathcal{R}, \mathcal{R}')$ must obey, since $\not\models \rho \wedge \varsigma \supset \epsilon_i$.
Let us address the consistency issue of $\Gamma \oplus_{\rangle\mathcal{R}'\langle} \mathcal{R}$. We have shown that $\{\gamma_1 \supset \gamma_2 \text{ s.t. } (\gamma_1, \emptyset, \gamma_2) \in \Gamma \setminus (\mathcal{R}, \mathcal{R}')\} \not\models \rho \wedge \varsigma \supset \epsilon_i$. Thus, $\{\gamma_1 \supset \gamma_2 \text{ s.t. } (\gamma_1, \emptyset, \gamma_2) \in \Gamma \setminus (\mathcal{R}, \mathcal{R}')\} \not\models \perp$. But, if a derivation of $(\top, \emptyset, \perp)$ from $\Gamma \setminus (\mathcal{R}, \mathcal{R}')$ were to exist, then it would contain positive nodes only (item 5 of Definition 3), which, due to item 4 of the aforementioned definition, would entail $\{\gamma_1 \supset \gamma_2 \text{ s.t. } (\gamma_1, \emptyset, \gamma_2) \in \Gamma \setminus (\mathcal{R}, \mathcal{R}')\} \models \perp$, which we just have proved to be impossible. Thus, there does not exist any derivation of $(\top, \emptyset, \perp)$ from $\Gamma \setminus (\mathcal{R}, \mathcal{R}')$. Accordingly, there does not exist any derivation of $(\top, \emptyset, \perp)$ from $\Gamma \setminus (\mathcal{R}, \mathcal{R}') \cup \{\mathcal{R}\}$ because such a derivation would have $\vdash \perp$ as its a root, which would violate item 2 from Definition 3 for all ϵ_i of $\mathcal{R}$ (there exists at least one since $n \geq 1$).
Let us finish the proof by showing that $\Gamma \oplus_{\rangle\mathcal{R}'\langle} \mathcal{R} \mid\!\not\sim^{\epsilon} \mathcal{R}'$. Since $\mathcal{R}'$ is a prime implicant of $\mathcal{R}$ modulo Γ, Definition 11 can apply and $\Gamma \oplus_{\rangle\mathcal{R}'\langle} \mathcal{R} = \Gamma \setminus (\mathcal{R}, \mathcal{R}') \cup \{\mathcal{R}\}$. But $\setminus(\mathcal{R}, \mathcal{R}')$ satisfies the constraint $\Gamma \setminus (\mathcal{R}, \mathcal{R}') \mid\!\not\sim^{\epsilon}_{\mathcal{R}} \mathcal{R}'$. In accordance with Definition 5, there does not exist any $\mathcal{R}$-derivation of $\mathcal{R}'$ from $\Gamma \setminus (\mathcal{R}, \mathcal{R}')$. Thus, there does not exist any derivation of $\mathcal{R}'$ from $\Gamma \setminus (\mathcal{R}, \mathcal{R}') \cup \{\mathcal{R}\}$. This means $\Gamma \oplus_{\rangle\mathcal{R}'\langle} \mathcal{R} \mid\!\not\sim^{\epsilon} \mathcal{R}'$. $\square$

The next step consists in iterating the above process on all primes implicants of $\mathcal{R}$. Assuming that the $\setminus$ operator is extended so that it applies to all the elements of its second argument, which is now a set of PEC rules; we only need one more definition.

Let $\mathcal{Y}$ be the set of prime implicants of $\mathcal{R}$ modulo Γ.

Definition 12. $\Gamma \oplus_{\rangle\mathcal{Y}\langle} \mathcal{R} =_{def} \Gamma \setminus (\mathcal{R}, \mathcal{Y}) \cup \{\mathcal{R}\}$.

It is easy to show that this definition enjoys the properties attached to the $\oplus_{\langle\mathcal{R}'\rangle}$ operator (see Theorem 1). Accordingly, the set of PEC rules delivered by $\Gamma \oplus_{\langle\mathcal{Y}\rangle} \mathcal{R}$ is consistent; it allows $\mathcal{R}$ to be deduced, while, at the same time, it prevents $\mathcal{R}$ to be subsumed.

Theorem 2. *Let Γ be a set of PEC rules. Let $\mathcal{R} = (\rho, \{\epsilon_1, \ldots, \epsilon_n\}, \varsigma)$ be a PEC rule s.t. $n \geq 1$ and s.t. $\epsilon_i \notin Cn(\{\rho, \varsigma\})$ for $i = 1..n$. Let $\mathcal{Y}$ be the set of prime implicants of $\mathcal{R}$ modulo Γ.*

- $\Gamma \oplus_{\rangle\mathcal{Y}\langle} \mathcal{R}$ *is consistent,*
- $\Gamma \oplus_{\rangle\mathcal{Y}\langle} \mathcal{R} \mid\!\sim^{\epsilon} \mathcal{R}$,
- $\Gamma \oplus_{\rangle\mathcal{Y}\langle} \mathcal{R} \mid\!\not\sim^{\epsilon} \mathcal{R}', \forall \mathcal{R}' \in \mathcal{Y}$.

10. Conclusions and future work

The contribution of this paper is at least twofold. First, a unified framework has been presented that allows both monotonic knowledge and defeasible rules to be represented and reasoned about in a uniform way. Derivation tools have been defined allowing to reason and infer both kinds of knowledge indifferently. The next step will be to address algorithmic aspects of X-derivations and associated inference, within the propositional setting. Also, the X-derivation concept implements the possibility to state defeasible rules as extra assumptions, which are coming in addition to the defeasible character of rules with exceptions. We believe that this two-level form of hypothetical reasoning could be further explored and refined. Also, a whole family of forms of implicants could be devised for defeasible rules, depending on the actual form of reasoning that is modeled and on the intended actual epistemological roles of the involved exceptions, premises and conclusion. Second, this framework has been exploited to solve a specific problem in knowledge representation and reasoning that has not received much attention so far. Namely, how could new information override the relevant subsuming information which is currently available? We claim that such an issue should not be taken for granted. Indeed, in real life we do often get new knowledge that is logically weaker but that appears *more informative* than the previously recorded one, and should therefore be preferred.

Acknowledgements

Part of this work has been supported by the *Région Nord/Pas-de-Calais* and the EC through a FEDER grant. The authors thank the anonymous reviewers for their valuable comments and suggestions.

References

[1] C.E. Alchourrón, P. Gärdenfors, D. Makinson, On the logic of theory change: partial meet contraction and revision functions, J. Symb. Log. 50 (2) (1985) 510–530.
[2] P. Besnard, Éric Grégoire, Ramon, S., Preemption operators, in: The 20th European Conference on Artificial Intelligence (ECAI 2012), 2012, pp. 893–894.
[3] P. Besnard, É. Grégoire, S. Ramon, Enforcing logically weaker knowledge in classical logic, in: H. Xiong, W.B. Lee (Eds.), KSEM, Lecture Notes in Computer Science, vol. 7091, Springer, 2011, pp. 44–55.
[4] S. Brass, On the semantics of supernormal defaults, in: Thirteenth International Joint Conference on Artificial Intelligence (IJCAI'93), 1993, pp. 578–583.
[5] A. Fuhrmann, S.O. Hansson, A survey of multiple contractions, J. Logic Lang. Inform. 3 (1) (1994) 39–76.
[6] R. Reiter, A logic for default reasoning, Artif. Intell. 13 (1980) 81–132.